# CHEN3600 – Computer-Aided Chemical Engineering Spring 2012

# Chemical Engineering Department Course Project

**T.D. Placek Auburn University**

 **Course Project – Interim Report 2**

During the past week, you have received a number of sources of information and data concerning the topic of heatwork. This has included: (1) cone temperature charts (from Orton), (2) output from the cone\_calc program from Orton, (3) web articles concerning the Arrhenius equation, and heatwork theory. We are now in a position to be ready to relate the heatwork which is actually accomplished during a kiln firing to a standardized firing based on the cone data provided by Orton.

In other words, you should be seeking to establish a procedure whereby you can accept time-temperature data from a kiln firing and establish whether or not a particular heatwork occurred. In order to evaluate the various solutions submitted by students, one or two sets of data will be provided to you similar to the data provided for firing data characterization. The central idea behind your procedure will be to integrate the appropriate portion of the data in MATLAB. Quadrature is a numerical method used to find the area under the graph of a function, that is, to compute a definite integral.

This is most easily accomplished using the “quadl” function which is preferred over the simpler “quad” function.

q = quad(***fun***,a,b) tries to approximate the integral of function fun from a to b to within an error of 1e-6 using recursive adaptive Simpson quadrature. ***fun*** is a function handle. Limits a and b must be finite. The function y = fun(x) should accept a vector argument x and return a vector result y, the integrand evaluated at each element of x.

You can also pass additional parameters to quad or quadl by providing these after any required parameters (similar to fsolve). Check the documentation.

q = quad(fun,a,b,tol) uses an absolute error tolerance tol instead of the default which is 1.0e-6. Larger values of tol result in fewer function evaluations and faster computation, but less accurate results.

[q,fcnt] = quad(...) returns the number of function evaluations.

The function quadl may be more efficient with high accuracies and smooth integrands.

The list below contains information to help you determine which quadrature function in MATLAB to use:

The quad function may be most efficient for low accuracies with nonsmooth integrands.

The quad function may be most efficient for low accuracies with nonsmooth integrands.

The quadl function may be more efficient than quad at higher accuracies with smooth integrands.

Example

function y = myfun(x)

y = 1./(x.^3-2\*x-5);

Then pass @myfun, a function handle to myfun, to quad, along with the limits of integration, 0 to 2:

Q = quad(@myfun,0,2)

Q = -0.4605

Alternatively, you can pass the integrand to quad as an anonymous function handle F:

F = @(x)1./(x.^3-2\*x-5);

Q = quad(F,0,2);

The general firing schedule that will be provided will be for a cone 6 glaze firing. You may wish to work just with the cone 6 area or to generalize your work to include nearby or all cones situations. For example, your solution might just be able to confirm a properly fired cone 6 run (as well as underfired or overfired) or to be able to quantify what cone something was fired to (for example, only a cone 4 was reached although a cone 6 might have been attempted.) Your grade will reflect the degree of difficulty you undertake in your study.

In this report, you will be again submitting a technical memorandum with appropriate sections and subsections. Focus additional attention on the communication of factual information including using the required citation format. Also, be sure to provide graphs and other information such as tables that follow the required format.